

## R Veera BrahmaNandam - Day 4 - JavaScript – Hands on Tasks

### Problem 1:

**Assessment Goal:** Ensure learners understand responsiveness and screen adaptability.

**Hands-on Tasks:** Add viewport meta tag to the HTML page, Use media queries to:

Change background colour on mobile screen, adjust font size for smaller screens, convert navigation into vertical layout on mobile, Test the page using browser responsive mode

### **Expected Outcome:**

A webpage that looks different and readable on mobile and desktop screens.

### Complete Source Code:

#### Index.html:

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <!-- 1. Viewport meta tag -->

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Responsive Website</title>

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <!-- Nav bar -->

    <nav class="navbar">

        <a href="#">Home</a>
        <a href="#">About</a>
        <a href="#">Services</a>
        <a href="#">Contact</a>
    </nav>

    <section class="content">

        <h1>Responsive Design Demo</h1>
        <p>The website is responsible for viewport meta tag and
```

css media queries.

```
</p>  
</section>  
</body>  
</html>
```

**style.css:**

```
/* Desktop - Default */  
  
body {  
    margin: 0;  
    font-family: Arial, Helvetica, sans-serif;  
    background-color: #87f7fff6; /* Desktop Background */  
}  
  
/* Navigation - horizontal layout fro desktpt */  
  
.navbar {  
    display: flex;  
    justify-content: center;  
    background-color: #2c1872;  
}  
  
.navbar a {  
    font-size: 20px;  
    font-weight: bold;  
    color: white;  
    padding: 20px 30px;  
    text-decoration: none;  
}  
  
/* Content section */  
  
.content {  
    padding: 40px;  
    text-align: center;  
}  
  
.content h1 {  
    font-size: 36px; /* Desktop font size */
```

```
}

.content p {
    font-family: 'Times New Roman', Times, serif;
    font-weight: bold;
    font-size: 20px;
}

/* Media Queries for mobile */

@media (max-width: 768px) {

    /* Change backgr color in mobile */

    body {
        background-color: #dbeafe;
    }

    .content h1 {
        font-size: 20px; /* font size for mobile screens */
    }

    .content p {
        font-size: 16px;
    }

    /* Convert navigation into vertical layout in mobile */

    .navbar {
        flex-direction: column;
        align-items: center;
    }

    .navbar a {
        width: 100%;
        text-align: center;
        border-bottom: 1px solid #555;
    }
}
```

## Code Screenshot:

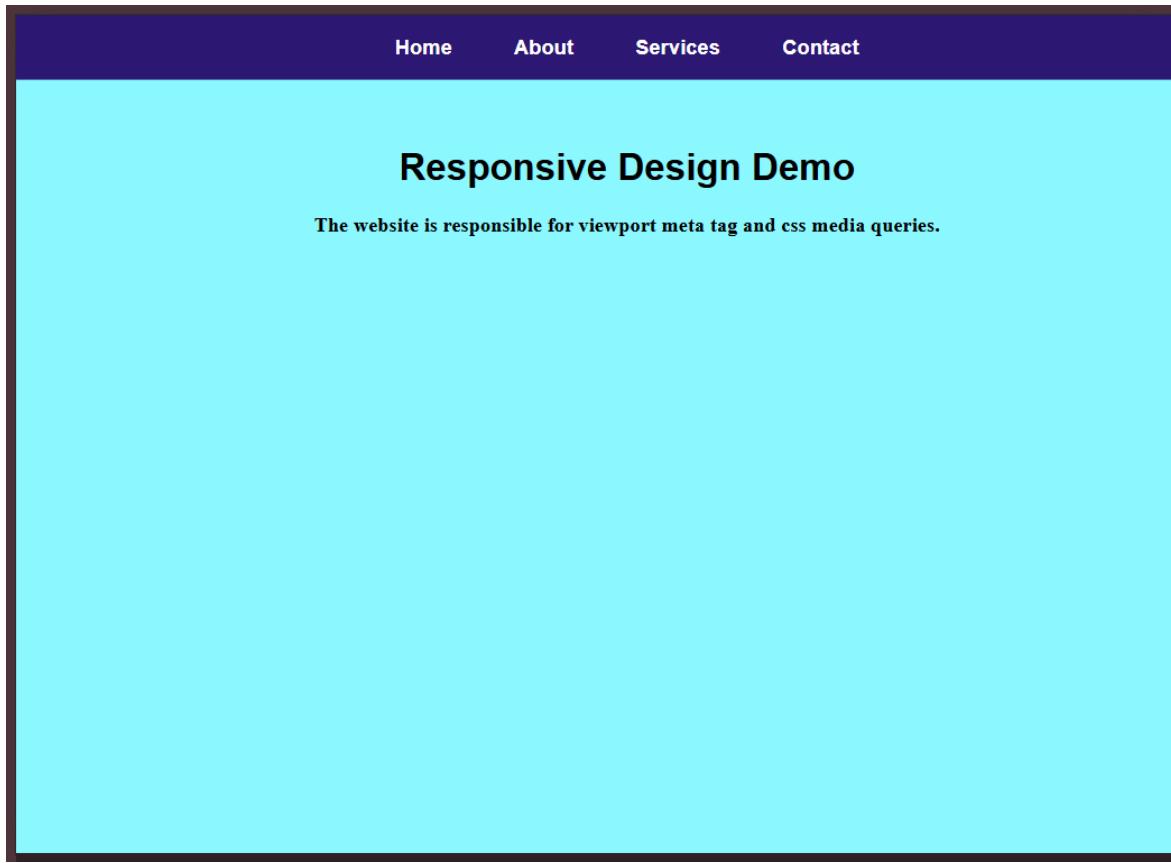
```
Task1 > index.html
1   <!-- R Veera BrahmaNadam - Day4 - JavaScript - Hands On Task
2
3   • Add viewport meta tag to the HTML page - Done
4   • Use media queries to:
5       • Change background color on mobile screen - Done
6       • Adjust font size for smaller screens - Done
7   • Convert navigation into vertical layout on mobile - Done
8   • Test the page using browser responsive mode - Done
9
10  -->
11
12  <!DOCTYPE html>
13  <html lang="en">
14  <head>
15      <meta charset="UTF-8">
16      <!-- 1. Viewport meta tag -->
17      <meta name="viewport" content="width=device-width, initial-scale=1.0">
18      <title>Responsive Website</title>
19      <link rel="stylesheet" href="style.css">
20  </head>
21  <body>
22
23      <!-- Nav bar -->
24      <nav class="navbar">
25          <a href="#">Home</a>
26          <a href="#">About</a>
27          <a href="#">Services</a>
28          <a href="#">Contact</a>
29      </nav>
30
31      <section class="content">
32          <h1>Responsive Design Demo</h1>
33          <p>The website is responsible for viewport meta tag and
34              |   css media queries.
35          </p>
36      </section>
37
38  </body>
39  </html>
40
```

```
Task1 > style.css
1  /* Desktop - Default */
2  body {
3      margin: 0;
4      font-family: Arial, Helvetica, sans-serif;
5      background-color: #87f7ffff; /* Desktop Background*/
6  }
7
8  /* Navigation - horizontal layout fro desktlp */
9  .navbar {
10      display: flex;
11      justify-content: center;
12      background-color: #2c1872;
13  }
14
```

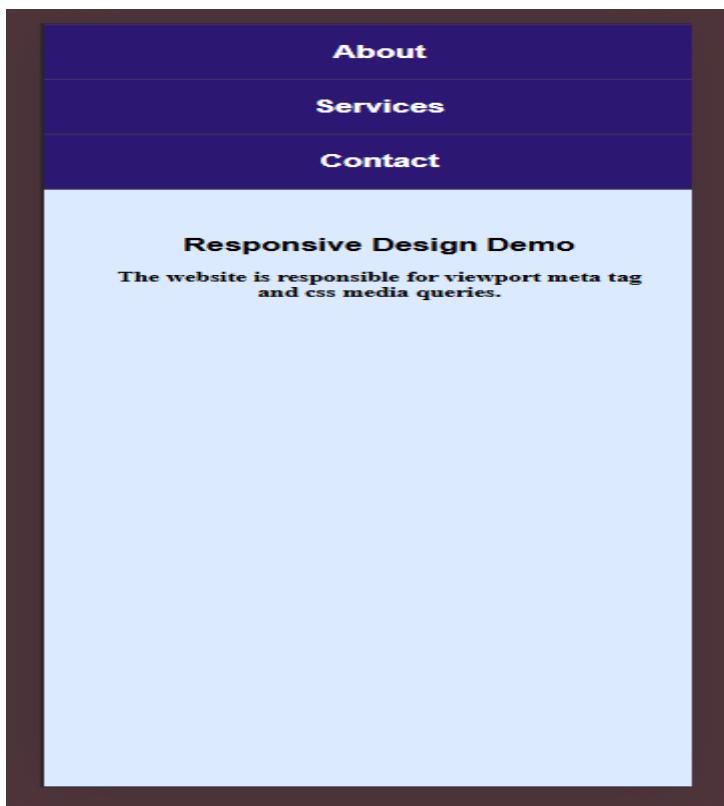
```
Task1 > # style.css > {} @media (max-width: 768px)
15   .navbar a {
16     font-size: 20px;
17     font-weight: bold;
18     color: black;
19     padding: 20px 30px;
20     text-decoration: none;
21   }
22
23 /* Content section */
24 .content {
25   padding: 40px;
26   text-align: center;
27 }
28
29 .content h1 {
30   font-size: 36px; /* Desktop font size */
31 }
32
33 .content p {
34   font-family: 'Times New Roman', Times, serif;
35   font-weight: bold;
36   font-size: 20px;
37 }
38
39 /* Media Queries for mobile */
40
41 @media (max-width: 768px) {
42
43   /* Change backgr color in mobile */
44   body {
45     background-color: black;
46   }
47
48   .content h1 {
49     font-size: 20px; /* font size for mobile screens */
50   }
51
52   .content p {
53     font-size: 16px;
54   }
55
56   /* Convert navigation into vertical layout in mobile */
57   .navbar {
58     flex-direction: column;
59     align-items: center;
60   }
61
62   .navbar a {
63     width: 100%;
64     text-align: center;
65     border-bottom: 1px solid black;
66   }
67 }
```

### Output Screenshot:

#### Laptop View:



Mobile View: iPhone 14 Pro Max



### **Code Explanation:**

- HTML uses the viewport meta tag so the page fits properly on mobile screens.
- HTML creates a simple structure using navigation and content sections.
- CSS styles the page of desktop view with proper colour and layout.
- CSS uses Flexbox to arrange the navigation links horizontally.
- Media queries are used in CSS to detect small screen sizes.
- On mobile screens, the background colour changes for better visibility.
- Font sizes are reduced on smaller screens to improve the readability.
- Navigation changes from horizontal to vertical layout on mobile.
- JavaScript can be used to add small interactive features if needed.
- The webpage looks clear and usable on both desktop and mobile screens.

### **Problem 2:**

#### **Assignment Goal: Student Grade Evaluator (Level-1)**

##### **Scenario**

A school wants a simple JavaScript program to evaluate a student's performance based on marks obtained in a subject.

❖ **Requirements:** Accept the student's marks as a variable, use if–else statements to assign grades:

- **Marks  $\geq$  75  $\rightarrow$  Grade A**
- **Marks  $\geq$  60  $\rightarrow$  Grade B**
- **Marks  $\geq$  40  $\rightarrow$  Grade C**
- **Marks  $<$  40  $\rightarrow$  Fail**

**Display the grade on the web page or console**

**Technical Constraints:** Use JavaScript variables (let or const), Use numeric data types, use comparison and logical operators, no functions or arrays allowed, Output using console.log() or document.write()

### **Completed Code:**

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Students Grade Evaluator</title>

    <link rel="stylesheet" href="">

</head>

<body>

    <h1>Students Grade</h1>

    <script>

        let marks = prompt("Enter your marks: ");

        if(isNaN(marks))

        {

            document.write("Invalid Input");

        }

        else if(marks >= 75)

        {

            document.write("Grade A");

        }

        else if(marks >= 60)

        {

            document.write("Grade B");

        }

        else if(marks >= 40)

        {

            document.write("Grade C");

        }

    </script>

</body>

</html>
```

```

else if(marks > 0 && marks <= 39)

{
    document.write("Fail");

}

else

{

    document.write("Negative Marks are not allowed for Students Unless any copying
happen 😊 😊 ");

}

</script>

</body>

</html>

```

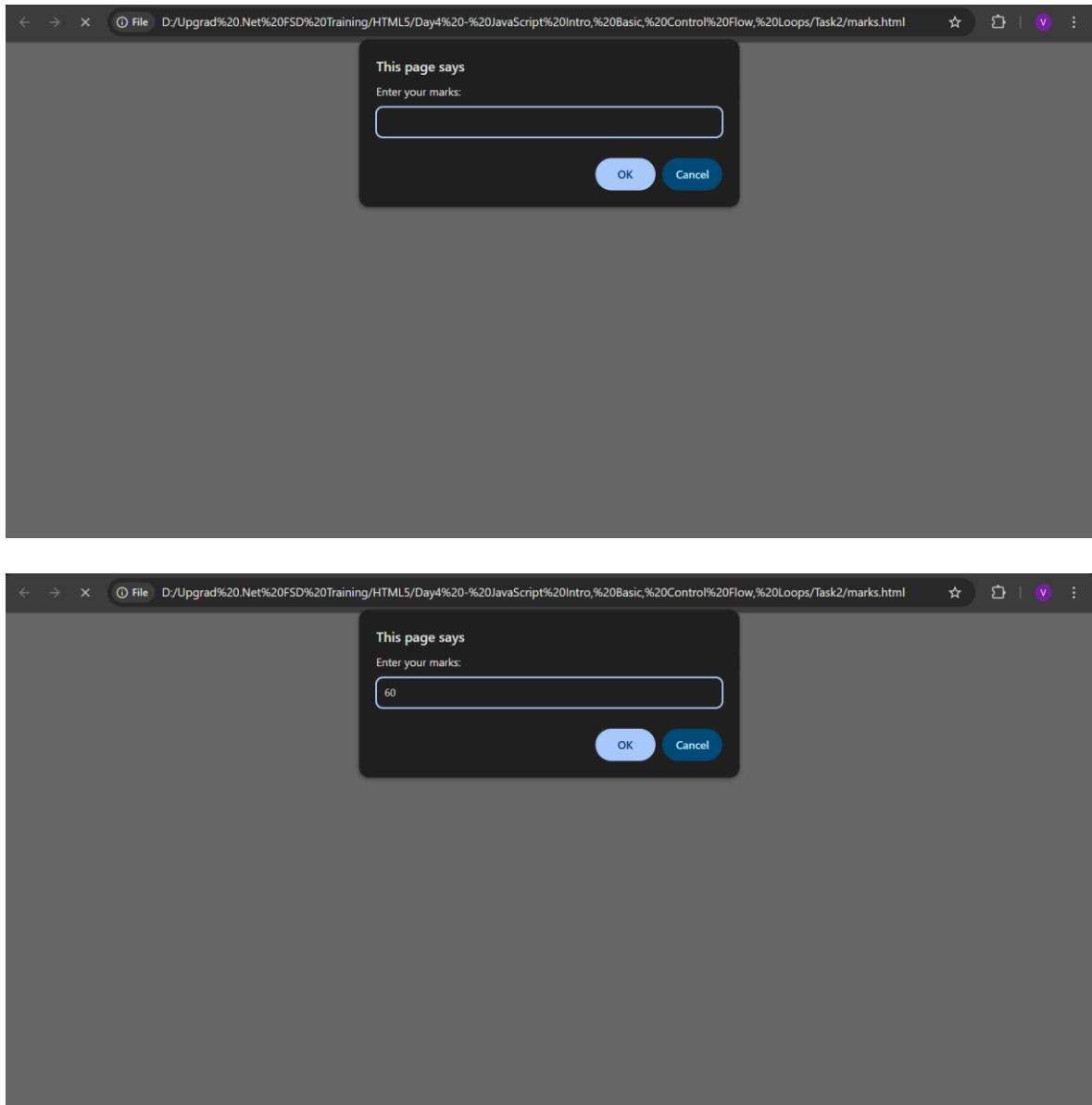
### Code ScreenShot:

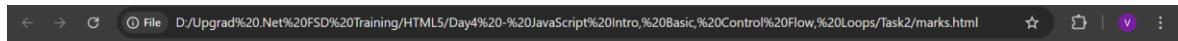
```

marks.html X Traffic_Signal.html NumberAnalysis.html index.html
Task2 > marks.html > html > body > script
13  <!DOCTYPE html>
14  <html lang="en">
15  <head>
16      <meta charset="UTF-8">
17      <meta name="viewport" content="width=device-width, initial-scale=1.0">
18      <title>Students Grade Evaluator</title>
19      <link rel="stylesheet" href="">
20  </head>
21  <body>
22      <h1>Students Grade</h1>
23
24      <script>
25
26          let marks = prompt("Enter your marks: ");
27
28          if(isNaN(marks))
29          {
30              document.write("Invalid Input");
31          }
32          else if(marks >= 75)
33          {
34              document.write("Grade A");
35          }
36          else if(marks >= 60)
37          {
38              document.write("Grade B");
39          }
40          else if(marks >= 40)
41          [
42              document.write("Grade C");
43          ]
44          else if(marks > 0 && marks <= 39)
45          {
46              document.write("Fail");
47          }
48          else
49          {
50              document.write("Negative Marks are not allowed for Students Unless any copying happen 😊 😊 ");
51          }
52
53      </script>
54  </body>
55  </html>

```

### Output Screenshot:





## Students Grade

Grade B

### Code Explanation:

- let marks = "A"; → Stores the student's marks in a variable
- isNaN(marks) → Checks whether the input is a character or non-numeric value
- console.log("Invalid input") → Displays message for invalid input
- if (marks >= 75) → Checks for Grade A condition
- else if (marks >= 60) → Checks for Grade B condition
- else if (marks >= 40) → Checks for Grade C condition
- else → Executes when marks are below 40
- >= → Comparison operator used for decision making
- if-else ladder → Controls the flow of conditions
- console.log() → Displays the result in the console

### Problem 3:

#### Simple Discount Calculator (Level-1)

##### Scenario

An online store wants to apply a discount based on the total purchase amount.

##### 📌 Requirements

- Store purchase amount in a variable
- Apply discount rules:
  - Amount  $\geq 5000 \rightarrow 20\% \text{ discount}$
  - Amount  $\geq 3000 \rightarrow 10\% \text{ discount}$
  - Amount  $< 3000 \rightarrow \text{No discount}$

- **Calculate and display:**
  - **Discount amount**
  - **Final payable amount**

### Technical Constraints

- **Use arithmetic operators**
- **Use if–else statements**
- **Use only primitive data types**

**No user input (hardcoded values allowed)**

#### **Complete Code:**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Simple Discount Calculator</title>
</head>
<body>
  <h1>Discount Calculator</h1>
  <script>
    let purchaseAmount = 4500;
    let discount = 0;
    let finalAmount = 0;
    if(purchaseAmount >= 5000)
    {
      discount = purchaseAmount * 0.20; // Arithmetic Operator
    }
    else if(purchaseAmount >= 3000)
    {
      // Amount >= 3000 ----> 10% discount
      discount = purchaseAmount * 0.10; // Arithmetic Operator Used
    }
    else
    {
```

```

// Amount < 3000 ---> No discount
discount = 0;
}

finalAmount = purchaseAmount - discount;

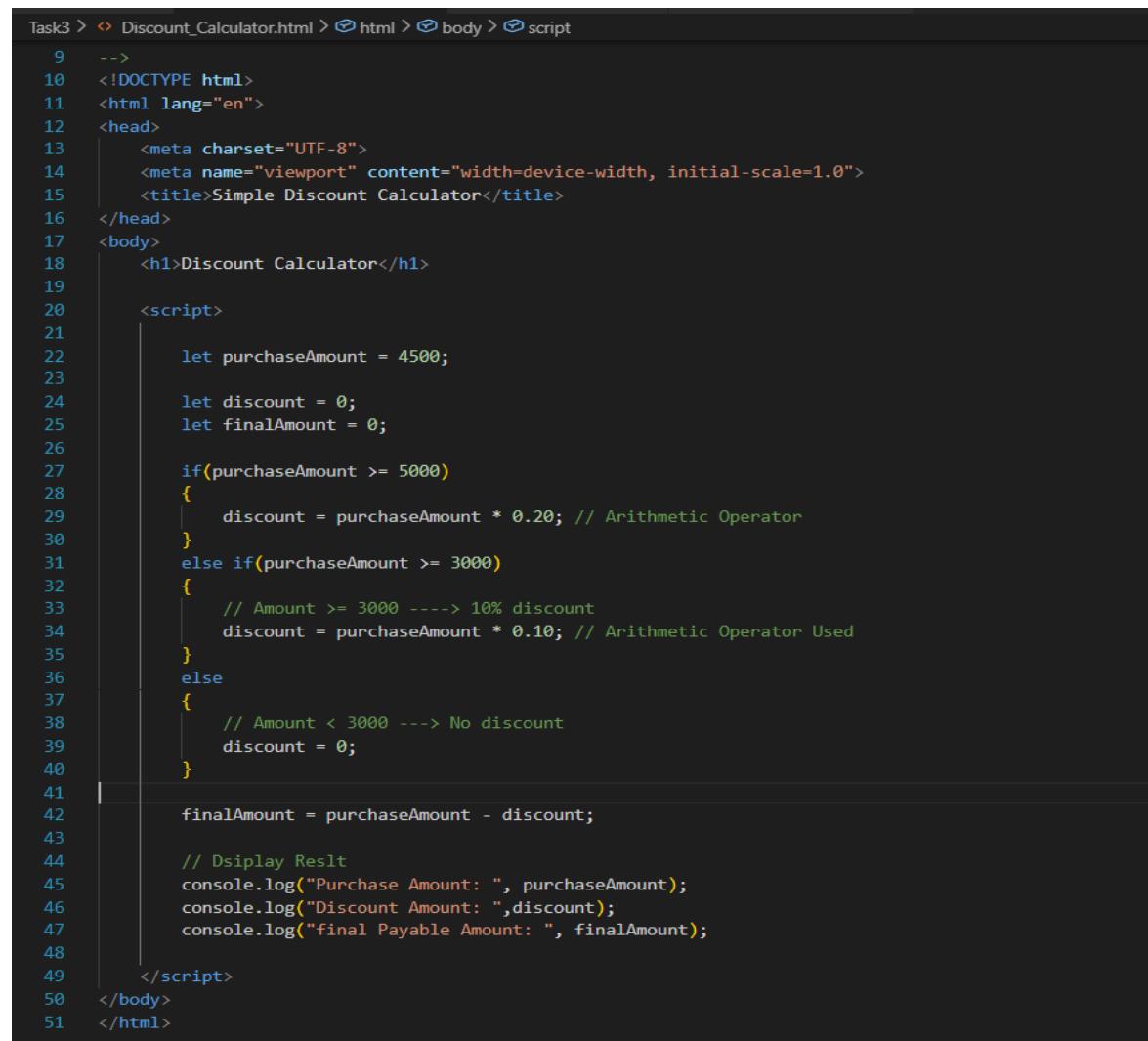
// Dsplay Reslt
console.log("Purchase Amount: ", purchaseAmount);
console.log("Discount Amount: ", discount);
console.log("final Payable Amount: ", finalAmount);

</script>

</body>
</html>

```

### Code Screenshot:

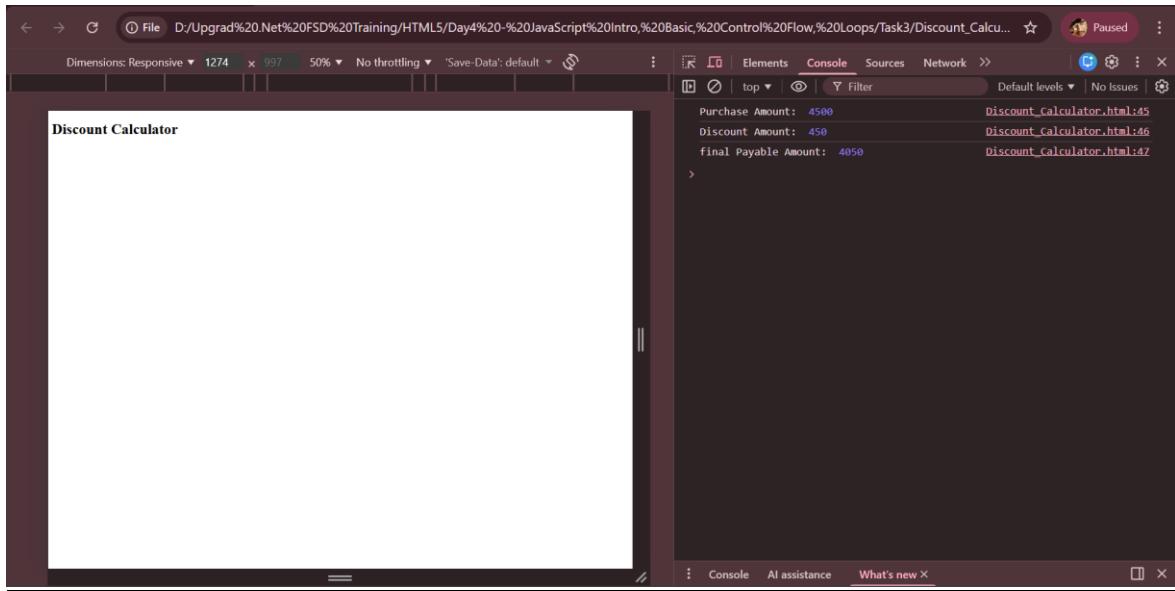


```

Task3 > Discount_Calculator.html > html > body > script
9   -->
10  <!DOCTYPE html>
11  <html lang="en">
12    <head>
13      <meta charset="UTF-8">
14      <meta name="viewport" content="width=device-width, initial-scale=1.0">
15      <title>Simple Discount Calculator</title>
16  </head>
17  <body>
18    <h1>Discount Calculator</h1>
19
20    <script>
21
22      let purchaseAmount = 4500;
23
24      let discount = 0;
25      let finalAmount = 0;
26
27      if(purchaseAmount >= 5000)
28      {
29          discount = purchaseAmount * 0.20; // Arithmetic Operator
30      }
31      else if(purchaseAmount >= 3000)
32      {
33          // Amount >= 3000 ----> 10% discount
34          discount = purchaseAmount * 0.10; // Arithmetic Operator Used
35      }
36      else
37      {
38          // Amount < 3000 ---> No discount
39          discount = 0;
40      }
41
42      finalAmount = purchaseAmount - discount;
43
44      // Dsplay Reslt
45      console.log("Purchase Amount: ", purchaseAmount);
46      console.log("Discount Amount: ", discount);
47      console.log("final Payable Amount: ", finalAmount);
48
49    </script>
50  </body>
51  </html>

```

### Output Screenshot:



### **Code Explanation:**

- A variable `purchaseAmount` is used to store the total purchase value.
- Only **primitive data types** (number) are used in the program.
- A variable `discount` is initialized to store the discount amount.
- An if–else statement checks the purchase amount.
- If the amount is **5000 or more**, a **20% discount** is calculated.
- If the amount is **3000 or more**, a **10% discount** is calculated.
- If the amount is **less than 3000**, no discount is applied.
- Arithmetic operators (\* and -) are used for calculations.
- The final payable amount is calculated by subtracting the discount.
- The purchase amount, discount, and final amount are displayed using `console.log()`.

### **Problem 4:**

#### **Traffic Signal Simulator (Level-2)**

##### **Scenario**

A traffic control system needs a JavaScript program that displays instructions based on traffic signal color.

##### **❖ Requirements**

- Store signal color in a variable ("red", "yellow", "green")
- Use a **switch statement** to display:
- Red → Stop

- Yellow → Get Ready
- Green → Go

Handle invalid signal input gracefully

## Technical Constraints

- Must use switch-case
- Use string data types
- Use console.log() for output
- No if-else allowed

### Complete Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Traffic Signal Simulator</title>
</head>
<body>
  <h1>Traffic Signal</h1>
  <script>
    // 1. Take Signal color from user
    let signal = prompt("Enter traffic signal color (red, yellow, green)");

    switch(signal)
    {
      case "red" :
      {
        console.log("Stop");
        break;
      }
      case "yellow" :
      {
        console.log("Get Ready");
        break;
      }
    }
  </script>
</body>
</html>
```

```

        case "green" :
    {
        console.log("Go")
        break;
    }
    default :
    {
        console.log("Invalid Signal. Pls enter Red, Yellow, or Green.");
    }
}
</script>
</body>
</html>

```

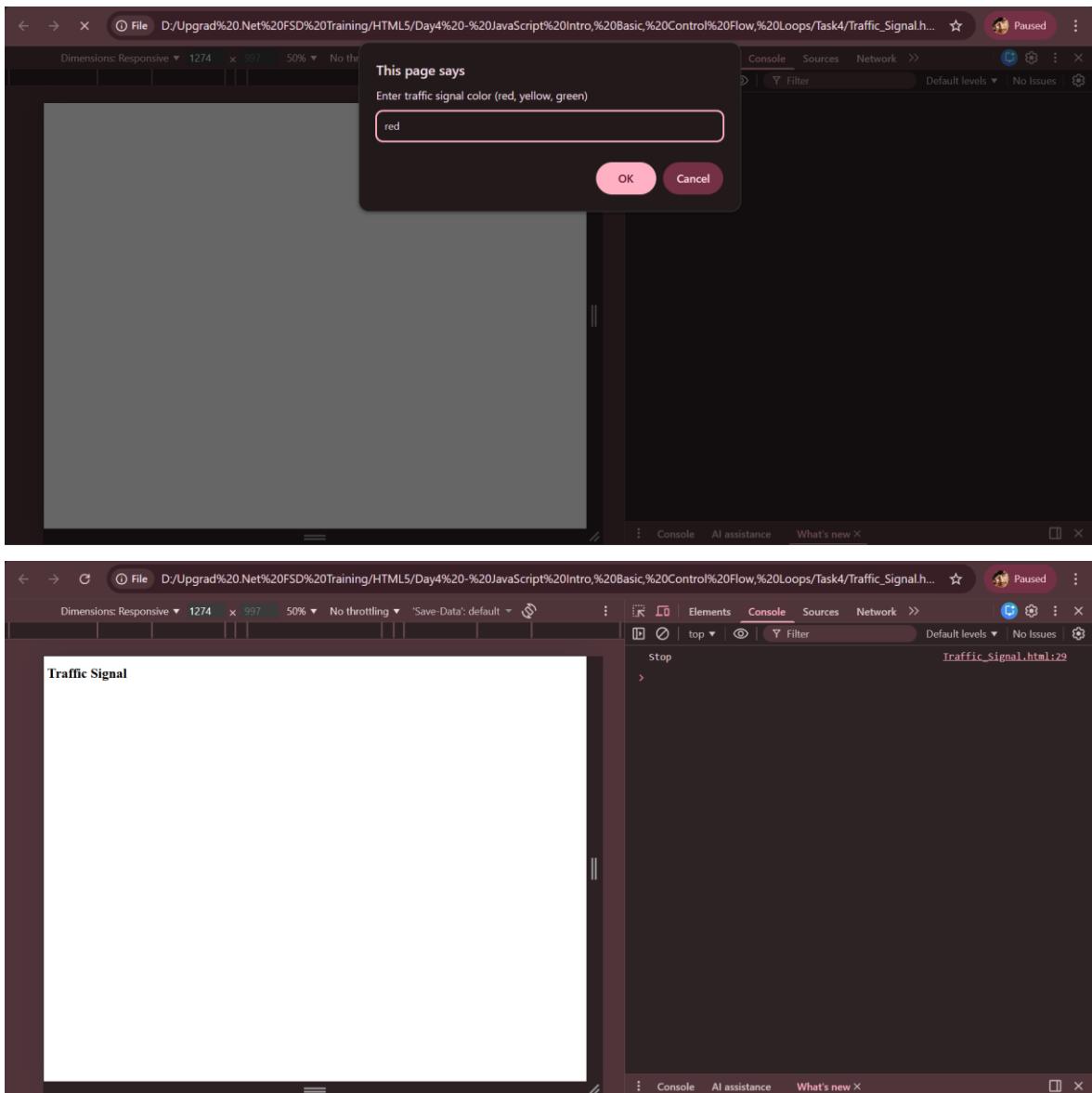
### Code Screenshot:

```

marks.html Discount_Calculator.html Traffic_Signal.html NumberAnalysis.html
Task4 > Traffic_Signal.html > html body script
9   <!DOCTYPE html>
10  <html lang="en">
11    <head>
12      <meta charset="UTF-8">
13      <meta name="viewport" content="width=device-width, initial-scale=1.0">
14      <title>Traffic Signal Simulator</title>
15    </head>
16    <body>
17
18      <h1>Traffic Signal</h1>
19
20      <script>
21
22          // 1. Take Signal color from user
23          let signal = prompt("Enter traffic signal color (red, yellow, green)");
24
25          switch(signal)
26          [
27              case "red" :
28              {
29                  console.log("Stop");
29                  break;
30              }
31
32              case "yellow" :
33              {
34                  console.log("Get Ready");
35                  break;
36              }
37
38              case "green" :
39              {
40                  console.log("Go")
41                  break;
42              }
43
44              default :
45              {
46                  console.log("Invalid Signal. Pls enter Red, Yellow, or Green.");
47              }
48          ]
49
50      </script>
51    </body>
52  </html>

```

### Output Screenshot:



### Code Explanation:

- `prompt()` is used to take the traffic signal color from the user.
- The input is stored in a variable `signalColor` as a string.
- `toLowerCase()` converts the input to lowercase to avoid case mismatch.
- A switch statement is used to control the program flow.
- The switch compares the user input with predefined string values.
- When the value is "red", the program prints Stop.
- When the value is "yellow", the program prints Get Ready.
- When the value is "green", the program prints Go.
- `break` is used after each case to stop further execution.
- The default case handles invalid inputs and shows an error message.

### Problem 5: Number Analysis Tool (Level-2)

## **Scenario**

A utility program is required to analyze numbers and provide insights such as positivity, parity, and range.

### Requirements

- Store a number in a variable
- Use **conditional (ternary) operator** to check:
- Positive or Negative
- Use **if–else** to check:
- Even or Odd
- Use a **loop** to print all numbers from 1 to the given number

### Technical Constraints

- Store a number in a variable
- Use conditional (ternary) operator to check:
- Positive or Negative
- Use if–else to check:
- Even or Odd
- Use a loop to print all numbers from 1 to the given number

### **Complete Code:**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Number Analysis Tool</title>
</head>
<body>
    <h1>Numbers</h1>
    <script>
        let number = prompt("Enter Number: ");
        let check = (number >= 0)? "Positive" : "Negative";
        console.log(number,"is a", check, "Number");
        if(number % 2 == 0)
    {
```

```

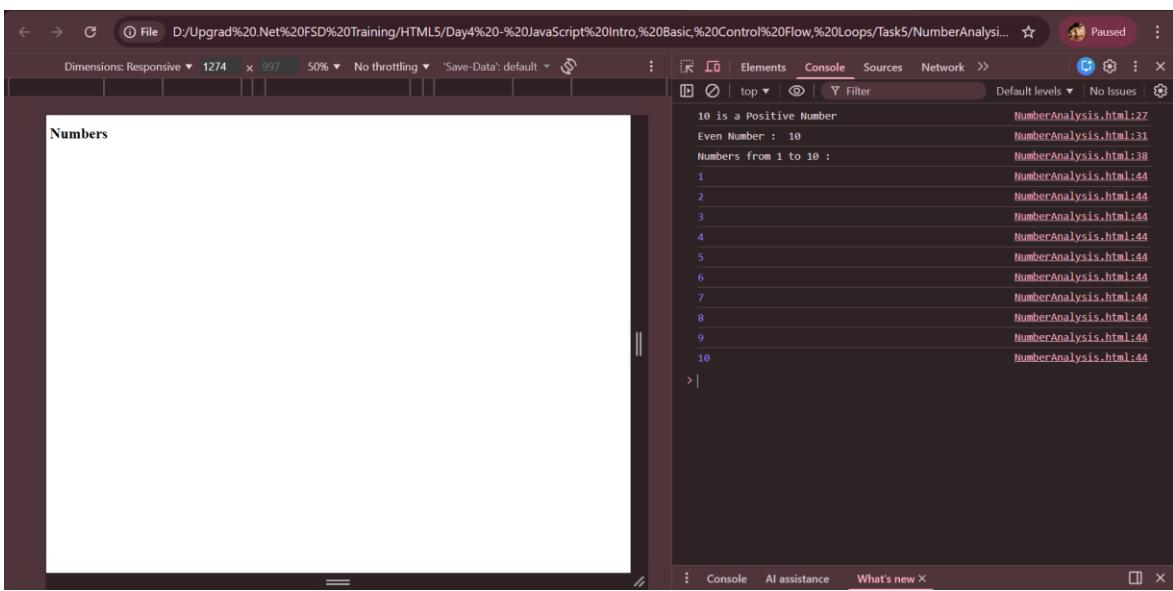
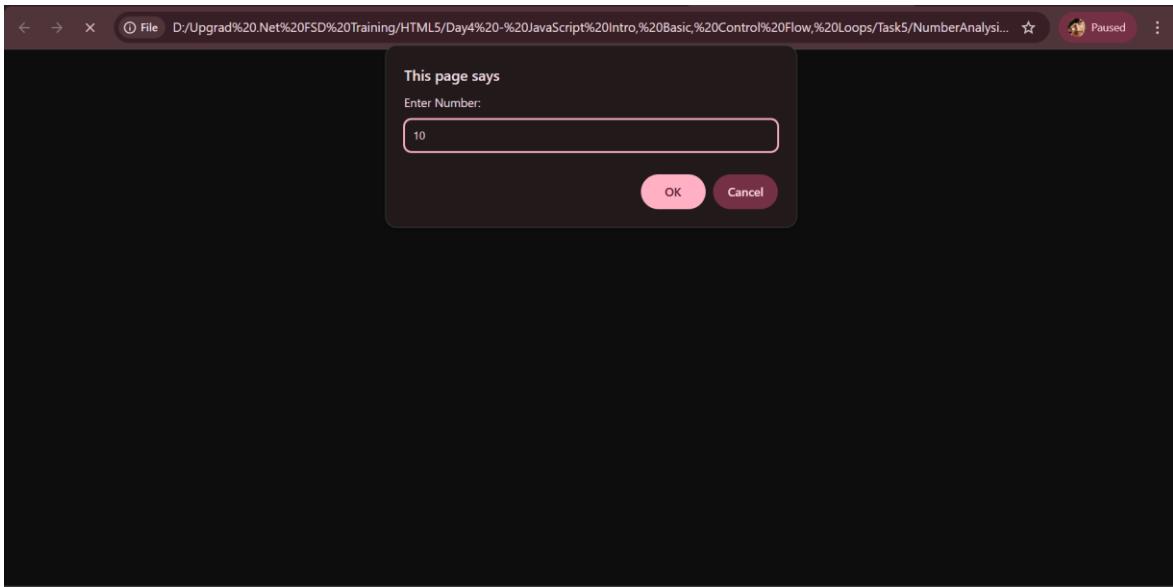
        console.log("Even Number : ",number);
    }
else
{
    console.log("Odd Number : ",number);
}
console.log("Numbers from 1 to " +number+ " :");
let i;
for(i=1; i<=number; i++)
{
    console.log(i);
}
</script>
</body>
</html>
```

### Code Screenshot:

```

10  <!DOCTYPE html>
11  <html lang="en">
12  <head>
13      <meta charset="UTF-8">
14      <meta name="viewport" content="width=device-width, initial-scale=1.0">
15      <title>Number Analysis Tool</title>
16  </head>
17  <body>
18
19      <h1>Numbers</h1>
20
21      <script>
22
23          let number = prompt("Enter Number: ");
24
25          let check = (number >= 0)? "Positive" : "Negative";
26
27          console.log(number,"is a", check, "Number");
28
29          if(number % 2 == 0)
30          {
31              console.log("Even Number : ",number);
32          }
33          else
34          {
35              console.log("Odd Number : ",number);
36          }
37
38          console.log("Numbers from 1 to " +number+ " :");
39
40          let i;
41
42          for(i=1; i<=number; i++)
43          {
44              console.log(i);
45          }
46
47      </script>
48
49  </body>
50  </html>
```

### Output Screenshot:



## Code Explanation:

- A variable number is declared to store the given numeric value.
  - The **ternary operator** checks whether the number is **positive or negative**.
  - The result of the ternary operation is stored in the variable positivity.
  - console.log() is used to display the positivity result.
  - The **modulus operator (%)** checks the remainder when dividing by 2.
  - An **if–else statement** determines whether the number is **even or odd**.
  - The even or odd result is printed using console.log().
  - A **for loop** is used to iterate starting from 1.
  - The loop runs until it reaches the given number.
  - Each number from 1 to the given number is printed during iteration.